

07/16/1999

E.I.C. Search History for Application No. 09/055,947

Source: Technical Non-Full Text Files on Dialog

- File 2:INSPEC 1969-1999/JUN W3
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS
- File 8:EI COMPENDEX(R) 1970-1999/JUN W3
(c) 1999 ENGINEERING INFO. INC.
- File 6:NTIS 64-1999/Jul W4
Comp&dist 1998 NTIS, Intl Copyright All Righ
- File 99:Wilson Appl. Sci & Tech Abs 1983-1999/May
(c) 1999 The HW Wilson Co.
- File 144:Pascal 1973-1999/May
(c) 1999 INIST/CNRS
- File 77:Conference Papers Index 1973-1999/Jul
(c) 1999 Cambridge Sci Abs
- File 434:Scisearch(R) Cited Ref Sci 1974-1989/Dec
(c) 1998 Inst for Sci Info
- File 34:SCISEARCH(R) CITED REF SCI 1990-1999/JUN W4
(c) 1999 INST FOR SCI INFO
- File 233:Microcomputer Abstracts 1974-1999/Jun
(c) 1999 Information Today Incl.
- File 238:Abs. in New Tech & Eng. 1981-1999/Jun
(c) 1999 Reed-Elsevier (UK) Ltd.
- File 65:Inside Conferences 1993-1999/May W5
(c) 1999 BLDSC all rts. reserv.
- File 94:JICST-EPLUS 1985-1999/MAR W3
(c)1999 JAPAN SCIENCE AND TECH CORP(JST)
- File 35:Dissertation Abstracts Online 1861-1999/Jun
(c) 1999 UMI

Set	Items	Description
S1	398729	(STATIC? OR PREDETERMIN? OR PRESET? OR PRE() (DETERMIN?
OR -		SET? ? OR SETTING))
S2	14956	INITIALI?
S3	400699	ARRAY?
S4	112170	(JAVA OR OBJECT()ORIENT? OR OOP OR OOPS OR
VIRTUAL()MACHINE		OR VM OR VMS OR JVM OR JVM5)
S5	34437	(CODE? ? OR CODING OR COMPILER?)(10N)(OPTIMI? OR REDUC?
OR		LESSEN? OR MINIMI? OR SHORTEN?)
S6	0	S1(10N)S2(10N)S3 AND S4 AND S5
S7	0	S1(S)S2(S)S3 AND (S4 OR S5)
S8	7	S1(5N)S2 AND S4
S9	7	RD (unique items)
S10	551478	(CLINIT OR CLASS OR (MULTI OR MULTIPLE OR MANY OR
NUMEROUS-)(2W)INSTRUCTION?)
S11	10663	(SINGLE OR ONE OR SOLE)(2N)(EXPRESSION OR INSTRUCTION
OR C-		OMMAND)
S12	1633354	REPLAC? OR SUBSTITUT? OR TAKE(1W)PLACE OR EXCHANG?
S13	0	S10(5N)S2 AND S12 AND S11 AND (S4 OR S5)
S14	0	S10(5N)S2 AND S12 AND (S4 OR S5)
S15	196	AU=(YELLIN, F? OR YELLIN F? OR TUCK, R? OR TUCK R?)
S16	0	S15 AND (S1(5N)S2 OR S4(S)S5 OR S12(S)S10)
S17	4	S15 AND S4
S18	3	RD (unique items)

E.I.C. Search History for Application No. 09/055,947

9/7/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS. All rts. reserv.

6249791 INSPEC Abstract Number: C1999-06-6110J-040
Title: Ensuring static initialization in C++
Author(s): Saks, D.
Journal: Embedded Systems Programming vol.12, no.3 p.109-11
Publisher: Miller Freeman,
Publication Date: March 1999 Country of Publication: USA
CODEN: EXPRE4 ISSN: 1040-3272
SICI: 1040-3272(199903)12:3L:109:ESI;1-R
Material Identity Number: O692-1999-005
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: C++ presents some hurdles to ensuring the
static
initialization of objects. The author gives some guidelines to help
you
work around them. The C++ standard does not obligate compilers to place
any
data into ROM. However, it leaves enough room so that compilers can
place
objects into ROM, provided that the objects have constant and
nonvolatile
types, occupy static storage and can be initialized statically
.
Compilers vary in their ability to initialize objects statically
.
However, the C++ standard specifies clear-cut circumstances in
which
compilers must use static initialization for an object, namely when
the
object occupies static storage, has only constant expressions in
its
initializer and has a POD (plain old data) type. Compilers do have
latitude
to treat certain dynamic initializations as static initializations
.
Thus, you may be able to place an object in ROM even if it has a
non-POD
type. On the other hand, you might find yourself writing
declarations in
C++ for constant, nonvolatile, statically allocated objects, only to
find
that the objects still don't make it into ROM. If that happens, look at
two
things: (1) make sure the initializer consists entirely of
constant
expressions (all it takes is one non-constant expression
in a
brace-enclosed list of initializers to turn a static
initialization
into a dynamic one); and (2) make sure the object has a POD type (if
it's a
class type, it might have bases or members that render it as a
non-POD
type). (0 Refs)

Copyright 1999, IEE

9/7/2 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS. All rts. reserv.

5790217 INSPEC Abstract Number: C9802-6140D-014
Title: Juggling with Java methods
Author(s): Hunt, J.
Journal: Application Development Advisor vol.1, no.1 p.40-2, 44-5
Publisher: SIGS,
Publication Date: Sept.-Oct. 1997 Country of Publication: UK
ISSN: 1369-4200
SICI: 1369-4200(199709/10)1:1L:40:JWJM;1-G
Material Identity Number: G363-97001
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: C and C++ developers often have trouble identifying how to use methods in Java . The author presents some tips and pitfalls to avoid. He considers naming methods, JavaBeans architecture, class-side methods, static initialisation blocks and constructors. (3 Refs)
Copyright 1998, IEE

9/7/3 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS. All rts. reserv.

5383826 INSPEC Abstract Number: C9611-6110J-019
Title: Being lazy about global-object initialization
Author(s): Downing, G.P.
Journal: Journal of Object-Oriented Programming vol.9, no.5 p.19-23
Publisher: SIGS Publications,
Publication Date: Sept. 1996 Country of Publication: USA
CODEN: JOOPEC ISSN: 0896-8438
SICI: 0896-8438(199609)9:5L:19:BLAG;1-G
Material Identity Number: N755-96007
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: In C++, implementation of global data as a class object can be accomplished in four ways: (1) using the global static object approach, (2) using the class static pointer with the implicit initialization approach, (3) using the class static pointer with the explicit initialization approach, and (4) using the class static pointer with the lazy initialization approach. Deciding which approach to use involves considering issues related to object construction, destruction,

initialization and performance. Each of the four approaches provides different tradeoffs in terms of these issues. (2 Refs)
Copyright 1996, IEE

9/7/4 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS. All rts. reserv.

04342258 INSPEC Abstract Number: C9303-6140D-067
Title: Static initializers: reducing the value-added tax on programs
Author(s): Reiser, J.F.
Author Affiliation: Mentor Graphics Corp., Wilsonville, OR, USA
Conference Title: USENIX C++ Technical Conference Proceedings p.171-80
Publisher: USENIX Assoc, Berkeley, CA, USA
Publication Date: 1992 Country of Publication: USA 347 pp.
Conference Date: 10-13 Aug. 1992 Conference Location: Portland, OR, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The declaration of a C++ initialised static object in a.h header file to provide initialization via 'allocation hook' acts much like a value-added tax: everyone pays, up and down the line. The software writer, the module integrator, and the maintenance engineer all pay. Even the end user of the software pays: invoking and terminating large systems can take dozens of seconds solely because of allocation hook static objects. The paper relates experience dealing with static initialization in several large software applications (thousands of classes, tens of thousands of functions, millions of instructions). Poor locality of instruction pages (large working set size) accounts for the runtime cost of static initialization . A facility for address tracing in real time, with on-line interactive graphical visualization, aids the exploration of how to reduce such costs. (11 Refs)

9/7/5 (Item 5 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS. All rts. reserv.

04342247 INSPEC Abstract Number: C9303-6140D-061
Title: USENIX C++ Technical Conference Proceedings
Publisher: USENIX Assoc, Berkeley, CA, USA
Publication Date: 1992 Country of Publication: USA 347 pp.

Conference Date: 10-13 Aug. 1992 Conference Location: Portland, OR, USA

Language: English Document Type: Conference Proceedings (CP)
Treatment: Practical (P)
Abstract: The following topics were dealt with: smart pointers; garbage collection and run-time typing as C++ library; encapsulating a C++ library; Sniff; C++ programming environment; statically typed abstract representation for C++ programs; CCEL; metalanguage for C++; space-efficient trees in C++; high performance scientific computing using C++; O-R gateway; system for connecting C++ application programs and static initializers ; Cdiff; syntax directed differencer for C++ programs; adding concurrency to a programming language; portable exception handling; assertion mechanism based on exceptions; communication facility for distributed object -oriented applications; client server application in C++; integrating Sun Microsystems XDR/RPC protocols into C++ stream model; run time type identification for C++; and run time information and class design.

9/7/6 (Item 1 from file: 233)
DIALOG(R)File 233:Microcomputer Abstracts
(c) 1999 Information Today Incl. All rts. reserv.

00444574 96IV12-004
Artificial life programming in Java -- Introduction to artificial life
O'Brien, Larry
InterActivity , December 1, 1996 , v2 nl3 p47-49, 3 Page(s)
ISSN: 1077-8047
Focuses on the meaning of artificial life and demonstrates its concepts and techniques through the discussion of an applet on flocking, schooling, and herding behavior written by Craig Reynolds in 1987. Says that the applet allows you to view specific behaviors such as herding and visual discrimination based on three essential rules: objects move toward a perceived center, an object matches its speed to the others around it, and an object must maintain a minimum distance between itself and other objects. Also discusses static variables, field initialization , and

thread programming. Includes one photo. (phi)

9/7/7 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abstracts Online
(c) 1999 UMI. All rts. reserv.

01609281 ORDER NO: AAD98-07703
FOXBOX: A SYSTEM FOR MANIPULATING SYMBOLIC OBJECTS IN BLACK BOX REPRESENTATION (PARALLEL COMPUTATION, OBJECT ORIENTED, EXPRESSION SWELL)
Author: DIAZ, ANGEL LUIS
Degree: PH.D.
Year: 1997
Corporate Source/Institution: RENSSELAER POLYTECHNIC INSTITUTE (0185)
Adviser: ERICH KALTOFEN
Source: VOLUME 58/09-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 4922. 98 PAGES

The dreaded phenomenon of expression swell in symbolic computation can be palliated by adopting implicit representations for symbolic objects, such as straight-line programs or so-called black box representations. In the latter, each expression is a symbolic object, more specifically, a computer program with a set of statically initialized data, which takes as input a value for each variable and then produces the value of the symbolic object it represents at the specified point.

In this thesis we introduce F sc OXB sc OX, a software package that puts in practice the black box representation of symbolic objects and provides algorithms for performing the symbolic calculus with such representations. Improved versions of the algorithms found in Kaltofen and

Trager (Journal of Symbolic Computing, vol. 9, nr. 3, p. 311 (1990)) and Kaltofen and Diaz (International Symposium on Symbolic and Algebraic Computation '95,) p. 232) are discussed. Also we describe an interpolation scheme based on a Zippel's algorithm (Journal of Symbolic Computing, vol. 9, nr. 3, p. 375 (1990)) that optimizes the number of required black box evaluations.

The design of F sc OXB sc OX is intended to demonstrate how plug-in software components can be employed for generally used symbolic systems. Our implementation incorporates data types parameterized by arbitrary coefficient domains and generic algorithms. By providing a mechanism for interfacing to general purpose computer algebra systems, we broaden F sc OXB scOX's applicability. Furthermore we provide a distribution mechanism that allows for parallel and distributed execution of F sc OXB sc OX

programs independent of the underlying parallel architecture.
Finally, we present the results of several challenge problems which exercise our F sc OXB sc OX library and represent the first symbolic solutions of such problems.
?ds
18/7/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1999 INSTITUTION OF ELECTRICAL ENGINEERS. All rts. reserv.

5486750 INSPEC Abstract Number: C9703-7430-004
Title: Inside the Java Virtual Machine
Author(s): Lindholm, T.; Yellin, F.
Journal: Unix Review vol.15, no.1 p.31-2, 34-6, 38-9
Publisher: Miller Freeman,
Publication Date: Jan. 1997 Country of Publication: USA
CODEN: UNRED5 ISSN: 0742-3136
SICI: 0742-3136(199701)15:1L:31:IJVM;1-L
Material Identity Number: G662-97001
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The Java Virtual Machine (JVM) is the run time module that executes the instructions encoded in Java bytecodes. As such, it serves as an interpreter of sorts whose design is conceptually simple. All Java programs are collections of classes. These classes are compiled into bytecodes that are run by the JVM . When invoked, the JVM loads the bytecode files and runs a verifier on them to ensure they are safe and properly formed; finally, the JVM steps through the file, executing instructions sequentially until the end of the program. The article discusses the JVM 's activities and aspects of the bytecodes themselves. To derive the most benefit from this discussion, you need some familiarity with the Java language and with the mechanics of program execution.
(0
Refs)
Copyright 1997, IEE

18/7/2 (Item 1 from file: 233)
DIALOG(R)File 233:Microcomputer Abstracts
(c) 1999 Information Today Incl. All rts. reserv.

00447838 97UR01-001
Inside the Java virtual machine -- Understanding the inner workings of the JVM will help you improve your Java application design
Lindholm, Tim; Yellin, Frank
UNIX Review , January 1, 1997 , v15 n1 p31-39, 7 Page(s)

ISSN: 0742-3136
Company Name: Sun Microsystems
Product Name: Java
Discusses the Java virtual machine (JVM) and various aspects of its byte codes. Explains that JVM is a run-time module that executes instructions encoded in Java byte codes. Adds that the Sun JVM implementation checks in four passes, studying questionable classes to verify if they satisfy necessary constraints at link time. Explains that primitive and reference values are the only two values that can be stored in variables, passed as arguments, returned as methods, and operated upon. Describes the verification process, run-time data areas, instruction set processing, and virtual machine exit. Includes one chart. (dpm)

18/7/3 (Item 1 from file: 65)
DIALOG(R)File 65:Inside Conferences 1993-1999/May W5
(c) 1999 BLDSC all rts. reserv. All rts. reserv.

01442681 INSIDE CONFERENCE ITEM ID: CN014316883
Low Level Security in Java
Yellin, F.
CONFERENCE: The web revolution-International world wide web conference;
4th (4th International world wide web conference)
WORLD WIDE WEB JOURNAL, 1995; 4th P: 369-380
O'Reilly & Associates, 1995
ISSN: 1085-2301 ISBN: 1565921690
LANGUAGE: English DOCUMENT TYPE: Conference Selected papers
CONFERENCE SPONSOR: Massachusetts Institute of Technology
Laboratory for Computer Science World Wide Web Consortium Team
Open Software Foundation Research Institute
CONFERENCE LOCATION: Boston, MA
CONFERENCE DATE: Dec 1995 (19951) (19951)

File 348:EUROPEAN PATENTS 1978-1999/JUN W25
(c) 1999 EUROPEAN PATENT OFFICE

Set	Items	Description
S1	187277	(STATIC? OR PREDETERMIN? OR PRESET? OR PRE() (DETERMIN?
OR -		SET? ? OR SETTING))
S2	19531	INITIALI?
S3	45190	ARRAY?
S4	3640	(JAVA OR OBJECT()ORIENT? OR OOP OR OOPS OR VIRTUAL()MACHINE
		OR VM OR VMS OR JVM OR JVMs)

E.I.C. Search History for Application No. 09/055,947

S5 4581 (CODE? ? OR CODING OR COMPILER?) (10N) (OPTIMI? OR REDUC?
OR
S6 0 S1(S)S2(S)S3(S)S4(S)S5
S7 1 S1(S)S2(S)S3(S) (S4 OR S5)
S8 21 S1(5N)S2(S)S3
S9 0 S8(S) (S4 OR S5)
S10 3 S1(5N)S2(S)S4
S11 30647 (CLINIT OR CLASS OR (MULTI OR MULTIPLE OR MANY OR
NUMEROUS-
) (2W) INSTRUCTION?)
S12 5908 (SINGLE OR ONE OR SOLE) (2N) (EXPRESSION OR INSTRUCTION
OR C-
OMMAND)
S13 233047 REPLAC? OR SUBSTITUT? OR TAKE(1W) PLACE OR EXCHANG?
S14 4 S11(S)S13(S)S1(S)S2
S15 0 S4(S)S5(S)S11(7N)S13
S16 0 SS S1(5N)S2(5N)S3
S17 0 AU=(YELLIN F? AND TUCK R?)
S18 9 AU=(YELLIN F? OR TUCK R?)
S19 1 S18 AND (S1(5N)S2 OR S4 OR S5)
?
10/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(C) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.

00837512
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
Network communications subsystem for networked digital computer system
Netzwerkkommunikationsuntersystem fur ein vernetztes
Digitalrechnersystem
Sous-systeme de communication de reseau pour un ordinateur lie a un
reseau
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View,
CA
94043, (US), (applicant designated states: DE;FR;GB;IT;NL)
INVENTOR:
Melo, Michael D., 12 Rainbow Lane, Billerica, Massachusetts 01821,
(US)
LEGAL REPRESENTATIVE:
Schmidt, Steffen J., Dipl.-Ing. (70552), Wuesthoff & Wuesthoff,
Patent-
und Rechtsanwalte, Schweigerstrasse 2, 81541 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 776111 A2 970528 (Basic)
APPLICATION (CC, No, Date): EP 96118790 961122;
PRIORITY (CC, No, Date): US 564381 951124
DESIGNATED STATES: DE; FR; GB; IT; NL
INTERNATIONAL PATENT CLASS: H04L-029/06; G06F-013/10;
ABSTRACT WORD COUNT: 282

LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) EPAB97 3552
SPEC A (English) EPAB97 8077
Total word count - document A 11629
9 of 53
07/16/99

E.I.C. Search History for Application No. 09/055,947

Total word count - document B 0
Total word count - documents A + B 11629
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
...SPECIFICATION at the beginning of a Windows session, Windows
notifies
drivers and other programs (generally "drivers") currently running in
the
computer system 10 that it is initializing by issuing a
predetermined
software interrupt, which is sequentially received and processed by
all
of the drivers in a software interrupt chain. The drivers which
receive
the notification may...
...Windows virtual device drivers" ("VxD's") that they may need to run
in
the Windows environment, which Windows may load and run in the system
virtual machine 32. With particular reference to the computer
system
10, the point-to-point protocol driver 25 may use the Windows
initialization notification to, in turn...

10/3,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(C) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.
00803537
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
OPERATING SYSTEM RUNTIME SUPPORTING RELEASE-TO-RELEASE BINARY
COMPATIBILITY
LAUFZEITSYSTEM EINES BETRIEBSSYSTEMS MIT UNTERSTUTZUNG
VON
BINARKOMPATIBILITAT ZWISCHEN VERSIONEN
VERSION MODIFIEE D'EXPLOITATION SUPPORTANT UNE COMPATIBILITE DE
VERSION A
VERSION AU NIVEAU BINAIRE
PATENT ASSIGNEE:
OBJECT TECHNOLOGY LICENSING CORP., (2168570), 10355 N. De Anza
Boulevard,
Cupertino, CA 95014, (US), (applicant designated states: DE;FR;GB)
INVENTOR:
HENINGER, Andrew, G., 1611 Corte Via, Los Altos, CA 94024, (US)
GIBBONS, Bill, 24737 Prospect Avenue, Los Altos Hills, CA 94022, (US)
MEYERS, Richard, 411 Esmeralda Drive, Santa Cruz, CA 95060, (US)
YEE, Terrence T., 18681 Maude Avenue, Saratoga, CA 95070, (US)
LEGAL REPRESENTATIVE:
Kindermann, Manfred (6412), Patentanwalt, Speerbergweg 29, 71032
Boblingen,
(DE)
PATENT (CC, No, Kind, Date): EP 811192 A1 971210 (Basic)
EP 811192 B1 990324
WO 9629648 960926
APPLICATION (CC, No, Date): EP 95939512 950929; WO 95US13077 950929
PRIORITY (CC, No, Date): US 406442 950320

E.I.C. Search History for Application No. 09/055,947

DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/44;
LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS B (English) 9911 1344
CLAIMS B (German) 9911 1270
CLAIMS B (French) 9911 1667
SPEC B (English) 9911 6109
Total word count - document A 0
Total word count - document B 10390
Total word count - documents A + B 10390

ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348

...SPECIFICATION release binary compatible class libraries to improve the execution speed and reduce the memory requirements of the computer program.
Vtables in traditional C++ implementations are initialized with static data. A problem confronting C++ runtime systems is that the size of virtual tables (vtables) can not be determined at compile time because the full...

...of a C++ runtime. US Patent 5,371,891 discusses an improved method and system for implementing constructors and destructors in a compiler for an object -oriented programming language is provided. In a preferred embodiment of the present invention, a construction displacement value is added to a pointer for a virtual function...

...occurrence of the base class within an instance of a derived class that inherits the base class. US Patent 5,339,438 discloses an interpretative object -oriented runtime environment.
US Patent 5,339,438 discloses a method, system and program for isolating the executable binary form of computer applications that use object...

10/3,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.
00297149
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
System for ensuring device compatibility.
System zur Geratkompatibilitatssicherung.
Systeme pour assurer la compatibilite d'un appareil.
PATENT ASSIGNEE:

E.I.C. Search History for Application No. 09/055,947

International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB)
INVENTOR:
Cicciarella, Raymond Jack, 401 Stark Avenue, Endwell New York 13760, (US)
Millis, David Burton, R.D. 1, Box 127, Friendsville Pennsylvania 18818, (US)
LEGAL REPRESENTATIVE:
Schafer, Wolfgang, Dipl.-Ing. (62021), IBM Deutschland Informationssysteme GmbH Patentwesen und Urheberrecht, D-70548 Stuttgart, (DE)
PATENT (CC, No, Kind, Date): EP 304866 A2 890301 (Basic)
EP 304866 A3 910306
EP 304866 B1 940216

APPLICATION (CC, No, Date): EP 88113694 880823;
PRIORITY (CC, No, Date): US 89200 870824
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-015/60; G06F-015/24; G06F-015/40;
ABSTRACT WORD COUNT: 129

LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS B (English) EPBBF1 389
CLAIMS B (German) EPBBF1 388
CLAIMS B (French) EPBBF1 433
SPEC B (English) EPBBF1 19457
Total word count - document A 0
Total word count - document B 20667
Total word count - documents A + B 20667

ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348

...SPECIFICATION relief types, and connector mating information. These items of information ensure that only information corresponding to predetermined requirements is used.
Other values that must be initialized include the information regarding the MDA system 15 to which cable files 18 will be sent. The particular address, the netid and other pertinent information for the hot reader 420 is a function within the VM operating system. It includes an auto-receive routine that allows the program 416 to update and add data
files 440 by means of sending request...
14/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.
00638555
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
Object system with derived metaclasses.
Objektsystem mit abgeleiteten Metaklassen.
Systeme objet avec metaclasses derivees.
PATENT ASSIGNEE:

E.I.C. Search History for Application No. 09/055,947

International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB)
INVENTOR:
Danforth, Scott Harrison, 10011 Woodland Village Drive, Austin, Texas 78750, (US)
LEGAL REPRESENTATIVE:
Burt, Roger James, Dr. (52152), IBM United Kingdom Limited Intellectual
Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 619544 A2 941012 (Basic)
EP 619544 A3 950201
APPLICATION (CC, No, Date): EP 94301931 940317;
PRIORITY (CC, No, Date): US 42930 930405
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/44;
ABSTRACT WORD COUNT: 141

LANGUAGE (Publication,Procedural,Application): English; English;
English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF2	472
SPEC A	(English)	EPABF2	15716
Total word count - document A			16188
Total word count - document B			0
Total word count - documents A + B			16188

ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348

...SPECIFICATION table is replaced by its redispatch stub. Redispatch stubs for inherited are determined by requesting them from the class' parent class.
Redispatch stubs for the class are generated by the SOM compiler and

supplied to the class initialization procedure in the calls to register each of the class ' static method. Control then passes to function block 1530 where the method procedure table entries for the class ' dispatch function are replaced' by the actual address of the class ' dispatch function (it is never correct to have a redispatch stub address in a dispatch function slot as this would result in a infinite loop...

14/3,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.

00546477
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
Computer with object oriented environment.
Rechner mit einer objektorientierten Umgebung.
Ordinateur avec un environnement oriente objet.

E.I.C. Search History for Application No. 09/055,947

PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states: CH;DE;FR;GB;IT;LI;NL;SE)
INVENTOR:
Conner, Mike H., 4416 Walhill Lane, Austin, Texas 78759, (US)
Martin, Andrew R., 1070 Mearns Meadow No. 534, Austin, Texas 78758, (US)
Raper, Larry K., 7860 Lakewood Drive, Austin, Texas 78750, (US)
LEGAL REPRESENTATIVE:
Burt, Roger James, Dr. (52152), IBM United Kingdom Limited Intellectual
Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 546809 A2 930616 (Basic)
EP 546809 A3 931118
APPLICATION (CC, No, Date): EP 92311207 921209;
PRIORITY (CC, No, Date): US 805779 911212
DESIGNATED STATES: CH; DE; FR; GB; IT; LI; NL; SE
INTERNATIONAL PATENT CLASS: G06F-009/44;
ABSTRACT WORD COUNT: 151

LANGUAGE (Publication,Procedural,Application): English; English;
English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	320
SPEC A	(English)	EPABF1	14070
Total word count - document A			14390
Total word count - document B			0
Total word count - documents A + B			14390

ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348

...SPECIFICATION the application program that is manipulating the object.

Method Procedure Table Initialization

Figure 15 is a flowchart depicting the detailed control flow that will properly initialize a method procedure table for a class that may change the association of method procedures to method during the execution of an application using the class . Control. commences at terminal block 1500 and immediately flows into function block 1510 where

space is allocated for the method procedure table. Enough space is allocated to contain an entry for the address of the class ' object and each of the method inherited or defined by the class in accordance with

Figure 7. Control then passes to function block 1520 where each method entry in the method procedure table is replaced by its redispatch stub.

Redispatch stubs for inherited are determined by requesting them from the class ' parent class . Redispatch stubs for the class are generated by the SOM compiler and supplied to the class initialization

procedure in the calls to register each of the class ' static method.
Control then passes to function block 1530 where the method procedure table entries for the class ' dispatch function are replaced by the actual address of the class ' dispatch function (it is never correct to have a redispatch stub address in a dispatch function slot as this would result in a infinite loop...

14/3,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.

00546431
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
Object oriented data processing system.
Objektorientiertes Datenverarbeitungssystem.
Systeme de traitement de donnees oriente objet.
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB)
INVENTOR:
Conner, Mike Haden, 4416 Walhill Lane, Austin, Texas 78759, (US)
Martin, Andrew Richard, 1070 Mearns Meadow No. 534, Austin, Texas 78758,
(US)
Raper, Larry Keith, 7860 Lakewood Drive, Austin, Texas 78750, (US)
LEGAL REPRESENTATIVE:
Burt, Roger James, Dr. (52152), IBM United Kingdom Limited
Intellectual
Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 546794 A2 930616 (Basic)
EP 546794 A3 931201
APPLICATION (CC, No, Date): EP 92311160 921208;
PRIORITY (CC, No, Date): US 805663 911212
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/44;
ABSTRACT WORD COUNT: 132

LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) EPABF1 448
SPEC A (English) EPABF1 14230
Total word count - document A 14678
Total word count - document B 0
Total word count - documents A + B 14678

ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348

....SPECIFICATION the application program that is manipulating the object.
Method Procedure Table Initialization

Figure 15 is a flowchart depicting the detailed control flow that will properly initialize a method procedure table for a class that may change the association of method procedures to method during the execution of an application using the class . Control commences at terminal block 1500 and immediately flows into function block 1510 where space is allocated for the method procedure table. Enough space is allocated to contain an entry for the address of the class ' object and each of the method inherited or defined by the class in accordance with
Figure 7. Control then passes to function block 1520 where each method entry in the method procedure table is replaced by its redispatch stub.
Redispatch stubs for inherited are determined by requesting them from the class ' parent class . Redispatch stubs for the class are generated by the SOM compiler and supplied to the class initialization procedure in the calls to register each of the class ' static method.
Control then passes to function block 1530 where the method procedure table entries for the class ' dispatch function are replaced by the actual address of the class ' dispatch function (it is never correct to have a redispatch stub address in a dispatch function slot as this would result in a infinite loop...

14/3,K/4 (Item 4 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.

00521707
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
Microcomputer and divider circuit.
Mikrocomputer und Teilerschaltung.
Microordinateur et circuit diviseur.
PATENT ASSIGNEE:
HITACHI, LTD., (204144), 6, Kanda Surugadai 4-chome, Chiyoda-ku, Tokyo
100, (JP), (applicant designated states: DE;FR;GB;IT)
HITACHI VLSI ENGINEERING CORPORATION, (762562), 5-20-1, Josuihon-cho, Kodaira-shi, Tokyo, (JP), (applicant designated states: DE;FR;GB;IT)
HITACHI MICROCOMPUTER SYSTEM LTD., (1298040), 5-22-1, Josuihon-cho, Kodaira-shi, Tokyo, (JP), (applicant designated states: DE;FR;GB;IT)
INVENTOR:
Kawasaki, Shumpei, 401, Gourudenmanshon, 4-19-6, Kamitakada, Nakano-ku, Tokyo, (JP)
Sakakibara, Eiji, 5-22-3, Josuihon-cho, Kodaira-shi, Tokyo, (JP)
Fukada, Kaoru, Shouburyo, 5-18-30, Midori-cho, Koganei-shi, Tokyo, (JP)
Yamazaki, Takanaga, B203, Gakuenso, 3-3-4, Gakuennishi-machi,

Kodaira-shi, Tokyo, (JP)
Akao, Yasushi, 4-15-9-505, Higashikoigakubo, Kokubunji-shi, Tokyo, (JP)
Baba, Shiro, 4-9-17, Tokura, Kokubunji-shi, Tokyo, (JP)
Kihara, Toshimasa, 7-2-6, Sunagawa-machi, Tachikawa-shi, Tokyo, (JP)
Kurakazu, Keiichi, 5003-12, Kamiyamaguchi, Tokorozawa-shi, Saitama, (JP)
Tsukamoto, Takashi, 101, Kkoup, 85-1, Naka-machi, Kodaira-shi, Tokyo, (JP)
Masumura, Shigeki, B-202, 3-3-7, Gakuennishi-machi, Kodaira-shi, Tokyo, (JP)
Tawara, Yasuhiro, Hitachiwakabahigashiryo, 5-22-3, Josuihon-cho, Kodaira-shi, Tokyo, (JP)
Kashiwagi, Yugo, 5-38-15-206, Hon-cho, Koganei-shi, Tokyo, (JP)
Fujita, Shuya, 1-26-13-312, Josuihon-cho, Kodaira-shi, Tokyo, (JP)
Ishida, Katsuhiko, Hitachikoganeiryo, 5-20-22, Midori-cho, Koganei-shi, Tokyo, (JP)
Sawa, Noriko, 4-27-1-301, Sekido, Tama-shi, Tokyo, (JP)
Asano, Yoichi, 4-11-17, Matsubara, Setagaya-ku, Tokyo, (JP)
Chaki, Hideaki, 2961-1-210, Wakasa-1-chome, Tokorozawa-shi, Saitama, (JP)
Sugawara, Tadahiko, 1-851-13, Ogawa-cho, Kodaira-shi, Tokyo, (JP)
Kainaga, Masahiro, 1-19-2, Utsukushigaoka, Midori-ku, Yokohama-shi, Kanagawa, (JP)
Noguchi, Kouki, B-11, Hitachitennoueshataku 1-217,, Higashikoigakubo Kokubunji-shi, Tokyo, (JP)
Watabe, Mitsuru, 1800-504, Ooaza Hirano Urizura, Naka-gun Ibaragi, (JP)
LEGAL REPRESENTATIVE:
Strehl, Schubel-Hopf, Groening (100941), Maximilianstrasse 54 Postfach 22
14 55, W-8000 Munchen 22, (DE)
PATENT (CC, No, Kind, Date): EP 525375 A2 930203 (Basic)
EP 525375 A3 930421
APPLICATION (CC, No, Date): EP 92110517 920622;
PRIORITY (CC, No, Date): JP 91178739 910624; JP 92154525 920521
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: G06F-009/38;
ABSTRACT WORD COUNT: 35

LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) EPABF1 2020
SPEC A (English) EPABF1 30571
Total word count - document A 32591
Total word count - document B 0
Total word count - documents A + B 32591
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
...SPECIFICATION processing. Specifically, the SR and the PC are released
to the stack, and a branch is made to an interruption routine in

accordance with a predetermined vector. The PC is the leading address
of the instruction immediately after the present instruction.
Operation: (see image in original document)
Example of Use: (see...
...is subtracted from the content of the general purpose register, and the
result is latched in the Rn. The subtraction with the immediate data is
replaced by ADD #imm,Rn.
Operation: (see image in original document)
Example of Use: (see image in original document)
SUBC (Subtraction with Carry) Instruction
Format:
SUBC... (Item 1 from file: 348)
19/5/1 File 348:EUROPEAN PATENTS
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 1999 EUROPEAN PATENT OFFICE. All rts. reserv.
01021345
ORDER fax of complete patent from Dialog SourceOne. See HELP ORDER 348
Method and apparatus for pre-processing and packaging class files
Verfahren und Gerat zur Vorverarbeitung und Verpackung von
Klassendateien
Procede et dispositif pour pretraitement et assemblage des fichiers
classes
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392737), 901 San Antonio Road, MS PAL01-521,
Palo Alto, California 94303, (US), (applicant designated states:
AT;BE;CH;CY;DE;DK;ES;FI;FR;GB;GR;IE;IT;LI;LU;MC;NL;PT;SE)
INVENTOR:
Fresko, Nedim, 1366 5th Avenue, Apt. 2, San Francisco, California 94122,
(US)
Tuck, Richard , 343 Hill Street, San Francisco, California 94114, (US)
LEGAL REPRESENTATIVE:
Goddar, Heinz J., Dr. (4231), FORRESTER & BOEHMERT Franz-Joseph-Strasse
38, 80801 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 913769 A2 990506 (Basic)
APPLICATION (CC, No, Date): EP 98120428 981028;
PRIORITY (CC, No, Date): US 961874 971031
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI;
LU; MC; NL; PT; SE
INTERNATIONAL PATENT CLASS: G06F-009/44;
ABSTRACT EP 913769 A2
A method and apparatus for pre-processing and packaging class files.
Embodiments remove duplicate information elements from a set of class files to reduce the size of individual class files and to prevent redundant resolution of the information elements. Memory allocation requirements are determined in advance for the set of classes as a whole

to reduce the complexity of memory allocation when the set of classes are loaded. The class files are stored in a single package for efficient storage, transfer and processing as a unit. In an embodiment, a pre-processor examines each class file in a set of class files to locate duplicate information in the form of redundant constants contained in a constant pool. The duplicate constant is placed in a separate shared table, and all occurrences of the constant are removed from the respective constant pools of the individual class files. During pre-processing, memory allocation requirements are determined for each class file, and used to determine a total allocation requirement for the set of class files. The shared table, the memory allocation requirements and the reduced class files are packaged as a unit in a multi-class file.

ABSTRACT WORD COUNT: 190
LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 990506 A2 Published application (Alwith Search Report ;A2without Search Report)
LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) 9918 1119
SPEC A (English) 9918 5322
Total word count - document A 6441
Total word count - document B 0
Total word count - documents A + B 6441

Source: European Patent File

T 14:49:16 ON 30 JUN 1999)

FILE 'COMPUAB, COMPUSCIENCE, ELCOM, INFODATA, SOLIDSTATE, CONF'
ENTERED
AT 14:49:35 ON 30 JUN 1999
16233 S STATIC? OR PREDETERMIN? OR PRESET? OR
PRE(W) (DETERMIN? OR SET
1439 S INITIALI?
L2 28490 S ARRAY?
L3 18510 S JAVA OR OBJECT ORIENT? OR OOP OR OOPS OR OOBEMS OR
L4 VIRTUAL MA
L5 1 S L1 AND L2 (7A) L3 AND L4
L6 2 S L1 (5A) L2 AND L4
L7 2 S L6 NOT L5
L8 2 DUPLICATE REMOVE L7 (0 DUPLICATES REMOVED)
L9 38082 S CLINIT OR CLASS OR (MULTI OR MULTIPLE OR MANY OR
NUMEROUS) (2W
L10 11 S L9 (5A) L2
L11 34275 S REPLAC? OR SUBSTITUT? OR EXCHANG? OR TAKE(1W) PLACE
L12 0 S L10 AND L11 AND L4

=> d 15 bib,abs

L5 ANSWER 1 OF 1 COMPUSCIENCE COPYRIGHT 1999 FIZ KARLSRUHE
AN 97(6):AC819 COMPUSCIENCE
TI Practical C++ programming.
AU Qualline, Steve
SO Sebastopol, CA: O ' Reilly and Associates, Inc. 1995. 557 p.
ISBN: 1-56592-139-9
DT Book
TC Theoretical
LA English
IP FIZKA
DN 9706-0420
AB This introductory C++ book assumes no knowledge of C. It has 29 chapters, divided into five parts (The Basics, '' Simple Programming, '' Advanced Types and Classes, '' Advanced Programming Concepts, '' and Other Language Features '').
The book
' s strong points include an admirable emphasis on style and documentation. It is easy to read, and the examples are easy to understand. Oualline discusses issues that arise when using a variety of compilers, including GNU C++, Turbo C++, Borland C++, and Visual C++.
Unfortunately, the book ' s weaknesses far outnumber its strengths. One problem is that 557 pages are not enough to achieve the author ' s goals: covering the C-like aspects of C++; describing the features that C++ adds to C; teaching structured programming and software engineering; discussing debugging, optimization, and portability; and explaining the nuances of floating-point arithmetic. The resulting book is broad but shallow. The level of discussion is uneven. Chapter 7 discusses what a directory is; chapter 9 uses 68000 assembly code to illustrate inline functions. The book ' s organization is similarly haphazard. A chapter on file I/O is followed by chapters on debugging and optimization, operator overloading, floating-point numbers, and pointers, in that order. The continue statement is covered in chapter 8, while the more common do statement and conditional operator are relegated to the next-to-last chapter, C++ ' s Dustier Corners. '' The book ' s C ancestry (it is a rewrite of an earlier book by the same author [1]) is apparent. The first significant C++ features do not show up until chapter 13, nearly 200 pages into the book. Even then, coverage of key features is skimpy.

Class derivation and virtual functions are described in a single 20-page chapter. Just 12 pages are devoted to templates; exceptions are covered in 10. There is no emphasis on full-fledged object-oriented programming; object-oriented analysis and design are barely mentioned. A language as complex as C++ cannot be explained properly without discussing nontrivial programs, yet nearly all of the programs in this book are small and artificial. The only substantial program appears in chapter 26, near the end of the book. The book ' s most annoying aspect is the sloppiness that is evident throughout. Typographical errors abound. Typefaces are used inconsistently. Programs contain variables that are never used; variables in a single example change names from page to page. Factual errors are not as common as typographical ones, but they do occur. The reserved word operator is used as a variable in chapter 7. The book claims that when an array parameter has more than one dimension, the lengths of all dimensions but the last must be specified. (In fact, the lengths of all but the first must be specified.) Other errors include an incorrect interpretation of too-short array initializers and a reference to the type long double as nonstandard. '' The book Algorithms + structures = programs {2} is credited to Knuth instead of Wirth. The book also suffers from inconsistent and confusing terminology: declarations, expressions, and \#define directives are all referred to as statements. '' Static is said to be an operator, cin and cout are classes, else is a statement, and storage classes are simply classes. Oversimplifications such as the preprocessor knows nothing about C++ are common. Some of the author ' s style dictums are ill-defined, arbitrary, and highly debatable. In his opinion, files should be limited to 1500 lines, and the main function should be less than two pages long (two sentences later, he limits the main function to three pages). The book ' s design is attractive, although many of the figures are overly ornate. Variables, for example, are shown as three-dimensional boxes

containing three-dimensional shaded digits. Reference material is minimal. Three appendices provide tables (ASCII characters, ranges of basic types, and operator precedence); the fourth contains a program that computes the sine function. More useful is the extensive glossary, which defines both C++ terms and various computer-related terms used in the book. Overall, I cannot recommend this book. There are much better treatments of C++ available, not the least of which is Stroustrup ' s own description of the language [3]. Readers who want a more gradual introduction to C++ would do better to start with a good C book, then make the transition to C++. [1] Oualline, S., Practical C programming., O ' Reilly, Sebastopol, CA, 1991. [2] Wirth, N., Algorithms + data structures = programs., Prentice-Hall, Englewood Cliffs, NJ, 1976. [3] Stroustrup, B., The C++ programming language, 2nd ed., Addison-Wesley, Reading, MA, 1991. => d l8 bib,abs 1-2

L8 ANSWER 1 OF 2 COMPUSCIENCE COPYRIGHT 1999 FIZ KARLSRUHE
AN 97(7):MA51343 COMPUSCIENCE
TI The Java language specification.
AU Gosling, James; Joy, Bill; Steele, Guy
SO Amsterdam: Addison Wesley. 1996. (xxv) 825 p.
ISBN: 0-201-63451-1
DT Book
TC Theoretical
CY Germany, Federal Republic of
LA English
IP FIZKA
DN 865.68001
AB Java is a general-purpose object-oriented language providing . concurrent behaviour. This issue, in a series of books dedicated to the language Java, presents a complete specification of this language and the main packages of its Application Programming Interface. The book contains 22 chapters. \par After the first chapter, an Introduction, the next one, called Grammars, informally presents the context-free grammars and their use in the specification of the lexical and syntactical structure of Java programs. \par Chapter 3, 'Lexical structure', specifies the basic lexical units of a Java program: white spaces, comments, identifiers,

keywords, literals, separators, operators as well as the Unicode used as the character set. \par Chapter 4, called 'Types, Values and Variables', presents the types used by the language, their set of values and different kinds of variables. Java is a strongly typed language, i.e. any expression has a type known at compile time. The types are divided into two groups: primitive types (boolean type, numeric type - integral and floating point -) and reference types (class type, interface type and array type). Also a special null type exists. The problem of variable initialization before use, is presented. \par Chapter 5, 'Conversions and Promotions', deals with specific conversions from a type to another. \par Chapter 6, 'Names', describes the name forms (simple and qualified), the declarations introducing names, the scope of a name. \par Chapter 7, 'Packages', introduces the structure of a Java program, the main program unit, called package, which is similar to modules in Modula or packages in Ada. The members of a package are the compilation units and the subpackages. \par Chapter 8, 'Classes', describes the Java's classes with their members: variables, methods, static initializers and constructors. The variables are divided into class variables and instance variables. The methods describe the code to be executed. Static initializers are parts of the code used to initialize a class. The constructors are special methods used to initialize new class instances. \par Chapter 9 describes Java's interface types. An interface declaration introduces a new reference type whose members are constants and abstract methods. \par Chapter 10 defines Java arrays. These are objects dynamically created and may be assigned to variables of type Object. \par Chapter 11 describes Java's exceptions. Similar to other languages (PL/I, Ada) Java takes into account the behaviour of a program violating the semantic constraints imposed by the language and provides an alternative continuation of the program. The 'catch' clause of 'try' statement explicitly handles the

exceptions occurring during the execution of a program. \par Chapter 12 presents the activities occurring during the execution of a Java program. Such a program is stored as binary files associated to classes and interfaces contained. The binary files are linked together with other classes and interfaces and then loaded into a Java Virtual Machine and executed. \par Chapter 13, 'Binary Compatibility', specifies the minimum standards for the binary compatibility guaranteed by all Java implementations. \par Chapter 14 deals with blocks and statements. Most of them are based on C and C++. Unlike C and C++, Java has no 'goto' statement, but provides 'break' and 'continue' statements with extensions allowing to specify statement labels. 'throw' and 'try' statements for exceptions and synchronized statements for achieving mutual exclusion, are specific only for Java. \par Chapter 15 specifies the meaning of Java expressions and the rules for their evaluation. \par Chapter 16 describes how Java ensures that local variables are definitely set before use, while other variables are automatically initialized to a default value. \par Chapter 17, 'Threads and Locks', describes the semantics of Java threads and locks. Each thread independently executes Java code. The synchronization of the threads is made by monitors. The behaviour of monitors is explained in terms of locks. \par Chapter 18 describes the possibilities of automatically generating documentation from special comments inserted in the source Java code. \par Chapter 19 presents a LALR(1) grammar for Java. The solutions adopted to transform the expository grammar into this LALR(1) grammar are described. \par Chapter 20 through 22 present a reference manual for the packages 'java.lang', 'java.util' and 'java.io' of the Java Application Programming Interface. \par The book ends with an index, credits for quotations used in this book and a colophon showing how the book was created. \par Java language is related to C and C++, but includes some concepts from other widespread languages (Modula, Ada). As the authors mention this language is intended

to be a production language rather than a research language. Now Java is a mature language and maybe with some modifications will be available for a widespread use. (M. Gheorghe (Bucuresti))

L8 ANSWER 2 OF 2 COMPUSCIENCE COPYRIGHT 1999 FIZ KARLSRUHE
AN 92(8):AC50541 COMPUSCIENCE
TI C++ for programmers.
AU Ammeraal, Leendert
SO Chichester, UK: John Wiley and Sons Ltd. 1991. 328 p.
Ser. Title: Wiley professional computing series.
ISBN: 0-471-93011-3
DT Book
IP ACM-CR
DN 9208-0541
AB Ten chapters make up this book: Introduction Expressions and Statements
More Operators Functions and Program Structure Arrays, Pointers, and Strings Classes and Objects Object-oriented Programming Standard I/O Stream I/O Standard Functions In addition, it has a table of contents, a preface, two appendices, a bibliography (short, as is usual in textbooks), and an index. This textbook with examples and exercises was written ``also{\ifmode\ldots\else\ldots\$}\fi\$for professional programmers who are already familiar with another language and want to switch to C++.'' It is supposed to be useful even to readers who do not know any programming language, but I do not think such readers can use this book for self-study. It feels more like a classroom textbook. A total of 58 exercises, most of them small and easy, accompany the chapters. Appendix A contains 47 further exercises; solutions to them are given at the end. Ammeraal's approach deviates from the conventional in that he requires no previous knowledge of C. On the contrary, ``newcomers in the C programming world should now seriously consider learning instead of C.'' This view of C++ as a better C pervades the book; in fact, the author also teaches some C as if it were an inferior C++. One goal was probably that the book should not frighten away potential readers by its sheer volume. In comparison, the new edition [1] of a textbook written by the father of C++ has 669 pages. Evidently, Ammeraal's book cannot fully cover all corners of the language, but the level of detail has been selected reasonably. The book is not

comprehensive enough to serve as the sole reference of a serious C++ programmer. The index could be more extensive for reference purposes. Ammeraal explains many important and difficult features of C++ well, such as addresses and pointers. He also warns against such ubiquitous pitfalls as pointers or references to deallocated objects at many relevant places in the book. The stress is on the conventional aspects of C++; the object-oriented features are presented briefly. Even this approach looks feasible: after all, to learn object-oriented programming one has to study more than a book on a particular language. In contrast to some other books, this volume has no artificially object-oriented examples in which almost everything is solved by inheritance. The downplaying of object orientation goes too far, unfortunately. Inheritance and virtual functions are presented with examples in chapter 7, but rather briefly. No exercises allow the reader to test his or her understanding of these important concepts. The style of the book is factual. The author has a positive attitude to the subject, but no exaggerated enthusiasm. I did not find any irony or puns in the text or examples. Many examples could have been improved by making them more interesting, more general, or both. The more complicated examples may need more explanation for readers studying independently. Almost all the exercises are mathematical problems. The author seems to be knowledgeable about his subject: I found few errors. The most surprising error was the repeated claim that the {\it struct} type constructor does not offer all the facilities that {\it class} has---this ceased to be true several C++ releases ago. The author makes a few minor erroneous claims: that an integer value can be assigned to an {\it enum} variable (p.\space 26); about the scope of a variable defined in a {\it for} statement (p.\space 37); and about the use of a stream variable as a truth value (p.\space 270). The explanation of the initialization of {\it static} (p.\space 83) and {\it external} (p.\space

84) objects is not completely correct. One requirement in exercise 7.4 (p.\space 227) is hardly possible to fulfill cleanly in C++; the given hint does not help with the problem. Chapter 10 is probably too optimistic about using the whole ANSI C standard library with C++ without any precautions. At least {\it setjmp} and {\it longjmp} can be dangerous. Ammeraal keeps to accepted C++ terminology except for one point: he systematically uses ``parameter'' to mean ``formal argument'' and ``argument'' to mean ``actual argument.'' The frequency of typographic errors seems to be below \mbox{average.} A large number of C++ textbooks are on the market today, most of which I have not seen. I will refrain from a comparative assessment, but teachers might well consider this book for students who do not know C, or if C++ is to be taught as the first programming language. [1] Stroustrup, B. The C++ programming language (2nd ed.) Addison-Wesley, Reading, MA, 1991.

Source: Technical Files on STN

File 351:DERWENT WPI 1963-1999/UD=9925;UP=9925;UM=9925
(c)1999 DERWENT INFO LTD
File 347:JAPIO Oct 1976-1999/Feb. (UPDATED 990603)
(c) 1999 JPO & JAPIO
File 344:Chinese Patents ABS Apr 1985-1999/Jun
(c) 1999 European Patent Office

Set	Items	Description
S1	353	(STATIC? OR PREDETERMIN? OR PRESET? OR PRE() (DETERMIN?
OR -		
S2	139062	SET? ? OR SETTING)) (5N)INITIALI?
S3	4949	ARRAY?
VIRTUAL()MACHINE		
S4	7034	(CODE? ? OR CODING OR COMPILER?) (7N) (OPTIMI? OR REDUC?
OR -		
S5		LESSEN? OR MINIMI? OR SHORTEN?)
S6	0	S1 AND S2 AND S3 AND S4
S7	0	S1 AND S2 AND S3
S8	0	S1 AND S3 AND S4
S8	1	S1 AND S3
S9	12	S1 AND S2
S10	688452	(STATIC? OR PREDETERMIN? OR PRESET? OR PRE() (DETERMIN?
OR -		
		SET? ? OR SETTING))
S11	98	S2(10N)INITIALI?

S12	6578	S3 OR MC=(T01-F07 OR T01-J20A OR T01-S01B)
S13	0	S11 AND S12
S14	5	S10(10N)INITIALI? AND S12
S15	21121	(CLINIT OR CLASS OR (MULTI OR MULTIPLE OR MANY OR NUMEROUS-
) (2W)INSTRUCTION?)
S16	5150	(SINGLE OR ONE OR SOLE) (2N) (EXPRESSION OR INSTRUCTION OR C-
		OMMAND)
S17	699821	REPLAC? OR SUBSTITUT? OR TAKE(1W)PLACE OR EXCHANG?
S18	0	S15 AND S17 AND S16 AND (S12 OR S1)
S19	28	S15(10N)INITIALI?
S20	0	S19 AND S17 AND (S12 OR S1)
S21	0	AU=(YELLIN F? AND TUCK R?)
S22	28	AU=(YELLIN F? OR TUCK R?)
S23	3	S22 AND (S1 OR S12)
	14/7/1	(Item 1 from file: 351)
DIALOG(R)File 351:DERWENT WPI		
(C)1999 DERWENT INFO LTD. All rts. reserv.		
012408239 **Image available**		
WPI Acc No: 99-214347/199918		
Object oriented component creating method		
Patent Assignee: SUN MICROSYSTEMS INC (SUNM)		
Inventor: FAUSTINI A		
Number of Countries: 001 Number of Patents: 001		
Patent Family:		
Patent No	Kind	Date Applicat No Kind Date Main IPC Week
US 5884078	A	19990316 US 97792245 A 19970131 G06F-009/44 199918 B

Priority Applications (No Type Date): US 97792245 A 19970131
Patent Details:
Patent Kind Ian Pg Filing Notes Application Patent
US 5884078 A 118

Abstract (Basic): US 5884078 A
NOVELTY - The component ports are initialized to a predetermined type and value, and polling them to ascertain whether and input has been coupled to one of the ports. The type and value of input is identified, if the input is coupled to the port. All the remaining ports are set to output the same type and value as that of input.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:
(a) object oriented component creating system;
(b) a computer program.
USE - In computer system.
ADVANTAGE - Enables the user to utilize object oriented component in an effective, non-intrusive manner.
DESCRIPTION OF DRAWING(S) - The figure shows flow chart of process of functioning of bicopy component.
Dwg.13A/25
Derwent Class: T01
International Patent Class (Main): G06F-009/44

E.I.C. Search History for Application No. 09/055,947

14/7/2 (Item 2 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1999 DERWENT INFO LTD. All rts. reserv.

011960217 **Image available**
WPI Acc No: 98-377127/199832
Static variable operation control for multi-threaded computer system - involves switching CPU to single thread mode to process static variable and then to multi-thread mode
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: KORENSHTEIN R; WANG I A
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Main IPC Week
US 5771382 A 19980623 US 95464355 A 19950605 G06F-009/44 199832 B
US 5771382 A 19980623 US 95464355 A 19950605 G06F-009/44 199832 B
US 97911187 A 19970814

Priority Applications (No Type Date): US 95464355 A 19950605; US 97911187 A 19970814

Patent Details:
Patent Kind Lan Pg Filing Notes Application Patent
US 5771382 A 15 Cont of US 95464355

Abstract (Basic): US 5771382 A
The variable operation control method involves receiving a lock request from a first thread to request CPU processing of at least one static variable operation. This is in response to a static variable initialization instruction. The CPU is switched to a single thread mode, where only the first thread is processed by the CPU during the single thread mode. The first thread is labelled after the step of switching the CPU to a single thread mode. This involves initializing a semaphore, where the semaphore indicates the lock request has been granted.
The CPU is switched to multiple thread mode, where multiple threads are processed by the CPU during multiple thread mode. A static variable initialization instruction is executed after switching the CPU to single thread mode and before switching it to multiple thread mode.
ADVANTAGE - Does not require compiler to change design, implementation or assumptions made regarding static initialisation .
Dwg.3/8

Derwent Class: T01
International Patent Class (Main): G06F-009/44
International Patent Class (Additional): G06F-009/46

14/7/3 (Item 3 from file: 351)
DIALOG(R)File 351:DERWENT WPI

E.I.C. Search History for Application No. 09/055,947

(c)1999 DERWENT INFO LTD. All rts. reserv.

011154160 **Image available**
WPI Acc No: 97-132084/199712
Equaliser control method for data communications - initialising data pointers and first buffer which is then filled with predetermined number of data samples with portion of data samples equalised in response
to enable signal generated cyclically by counter
Patent Assignee: TEXAS INSTR INC (TEXI)
Inventor: ANDREWS M S
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Main IPC Week
US 5602872 A 19970211 US 94359123 A 19941219 H03H-007/30 199712 B
US 5602872 A 19970211 US 94359123 A 19941219 H03H-007/30 199712 B

Priority Applications (No Type Date): US 94359123 A 19941219

Patent Details:
Patent Kind Lan Pg Filing Notes Application Patent
US 5602872 A 8

Abstract (Basic): US 5602872 A
The method involves initialising a number of data pointers and a first buffer stored in an input memory. A buffer full flag is generated in response to filling the first buffer with a predetermined number of data samples. A counter is initialised in response to the buffer full flag and an enable signal is generated every M cycles. A portion of the predetermined number of data samples is equalised. The portion is located using the data pointers, which are adjusted in response to a fractionally-spaced equaliser. The equaliser includes an adaptive filter, with the data transmitted through the data channel oversampled at an oversampling rate M.
ADVANTAGE - Efficient equaliser control with reduced ISI on oversampling transmitted data signal.
Dwg.2/8

Derwent Class: T01; U22; W01
International Patent Class (Main): H03H-007/30

14/7/4 (Item 4 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1999 DERWENT INFO LTD. All rts. reserv.

011088254 **Image available**
WPI Acc No: 97-066178/199707
Capacity symbol control method for exchange - transmitting system capacity characteristic codes from static and dynamic tables from exchange to terminal, and sending desired characteristic symbols to exchange to change status of exchange
Patent Assignee: SIEMENS AG (SIEI)

E.I.C. Search History for Application No. 09/055,947

access the methods without requiring externalisation of the method names.

The information provided by the specific procedures is retained by the class object and is accessible via class methods whenever the information is required. Any additional supporting information of methods, in particular the offset in the method procedure table for each method, is recorded in a single externally named data structure.

ADVANTAGE - Combination of two mechanisms eliminates requirement of external names on per method basis.

Dwg.9/15

Abstract (Equivalent): US 5361350 A

Management is accomplished by the operation of an algorithm in the memory of a processor which employs two mechanisms. First, the class method procedure tables are initialised by class specific procedures.

This allows applications to access the methods without requiring externalisation of the method means. The information provided by

the specific procedures is retained by the class object and is accessible via class methods whenever the information is required. Second, any additional supporting information for methods, in particular the offset in the method procedure table for each method, is recorded in a single externally data structure. The combination of

the two mechanisms eliminates the requirement of external names on a per method basis. USE - For effectively managing class method names by collecting representations of all of the names and additional supporting information in a single data structure.

Dwg.2/15

Derwent Class: T01

International Patent Class (Main): G06F-009/44; G06F-012/02; G06F-013/00

23/7/1 (Item 1 from file: 351)

DIALOG(R)File 351:DERWENT WPI

(c)1999 DERWENT INFO LTD. All rts. reserv.

012459557 **Image available**

WPI Acc No: 99-265665/19923

Preprocessing and packaging class files for object oriented computer applications

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)

Inventor: FRESKO N; TUCK R

Number of Countries: 025 Number of Patents: 001

Patent Family:

Patent No Kind Date Applicat No Kind Date Main IPC Week
EP 913769 A2 19990506 EP 98120428 A 19981028 G06F-009/44 199923 B

Priority Applications (No Type Date): US 97961874 A 19971031

33 of 53

07/16/99

E.I.C. Search History for Application No. 09/055,947

Patent Details:

Patent Kind Lan Pg Filing Notes Application Patent

EP 913769 A2 E 61

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE

IT

LI LT LU LV MC MK NL PT RO SE SI

Abstract (Basic): EP 913769 A

NOVELTY - A computer (200) receives downloaded class files and stores them as a single package. During preprocessing, memory allocation requirements are calculated for each file and duplicated elements in the files are separated into a shared table.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for a computer program product, for preprocessing and packaging apparatus and for a data storage medium.

USE - Preprocessing and packaging class files, specifically

object

-oriented computer applications; which are downloaded from the internet in platform independent format, e.g. Java (RTM) made up

of

class files, that can be assembled by a virtual machine process on

the required operating platform.

ADVANTAGE - Removes duplicated information elements from set of class files to reduce size of individual files.

DESCRIPTION OF DRAWING(S) - The drawing is a block diagram of a computer system implementing the preprocessing.

Bi-directional system bus 218

Central processor unit 213

Communication interface 220

Computer 200

Dwg.2/6

Derwent Class: T01

International Patent Class (Main): G06F-009/44

23/7/2 (Item 2 from file: 351)

DIALOG(R)File 351:DERWENT WPI

(c)1999 DERWENT INFO LTD. All rts. reserv.

011981600 **Image available**

WPI Acc No: 98-398510/199834

Program source code compilation method e.g. for computer - involves constructing file location identifier for identified symbol if symbol

is reference to remotely located file and then adding portion of fetched file to set of program codes

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)

Inventor: GOSLING J A; VAN HOFF A A; YELLIN F

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No Kind Date Applicat No Kind Date Main IPC Week
US 5778231 A 19980707 US 95575356 A 19951220 G06F-009/45 199834 B

Priority Applications (No Type Date): US 95575356 A 19951220

Patent Details:

Patent Kind Lan Pg Filing Notes Application Patent

34 of 53

07/16/99

US 5778231 A 9

Abstract (Basic): US 5778231 A
The method involves identifying each symbol in a source code that references another program. A judgment is performed to determine whether the identified symbol is a reference to a remotely located file or locally stored file. If the identified symbol is a reference to a remotely located file, a file location identifier is constructed for the symbol. A file location identifier based fetching process is performed.
A portion of the fetched file is added to a set of program codes to be compiled if the fetching is successfully performed. Otherwise the compilation of the source program is aborted. If the identified symbol is a reference to a locally stored file, file fetching is performed. A portion of the fetched file is added to the program codes.
ADVANTAGE - Allows for robust performance on computer network such as internet. Enables automatic recompilation of source code if compiled code is older than source code.
Dwg.1/2
Derwent Class: T01
International Patent Class (Main): G06F-009/45

23/7/3 (Item 3 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1999 DERWENT INFO LTD. All rts. reserv.

011374786 **Image available**
WPI Acc No: 97-352693/199733
Computer system storing several objects and procedures based on object class - determines if sole purpose of called method is to access private variable value and modify it or return constant, if so interpreter modifies method to directly access variable or constant using load and store instructions
Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: YELLIN F
Number of Countries: 012 Number of Patents: 007
Patent Family:
Patent No Kind Date Applicat No Kind Date Main IPC Week
EP 778521 A2 19970611 EP 96308721 A 19961203 G06F-009/45 199733 B
AU 9671889 A 19970612 AU 9671889 A 19961120 199733
CA 2191411 A 19970609 CA 2191411 A 19961127 G06F-009/45 199741
JP 10040107 A 19980213 JP 96328896 A 19961209 G06F-009/44 199817
US 5794044 A 19980811 US 95569754 A 19951208 G06F-009/45 199839
TW 341687 A 19981001 TW 96115368 A 19961212 G06F-009/06 199904
KR 97049812 A 19970729 KR 9662296 A 19961206 G06F-019/00 199909

Priority Applications (No Type Date): US 95569754 A 19951208
Patent Details:
Patent Kind Lan Pg Filing Notes Application Patent
EP 778521 A2 E 13
Designated States (Regional): DE FR GB IT NL SE
JP 10040107 A 14

Abstract (Basic): EP 778521 A
The computer system has secure program interpreter performs (114) a special check the first time it executes a method call to determine if the sole purpose of the called method is to access the value of the private variable and modify the value or return a constant value.
If so the interpreter modifies the method to directly access the variable or the constant. This modification uses privileged load and store instructions, not available in normal source code programs, that access private variables and constants outside the method being executed without causing security violation to be flagged.
When the modification is executed, the private variable or constant is accessed directly by the executed method using the privileged load and store instruction to avoid the flagging. When execution is complete the modification is flushed so that when the program is executed again the interpreter generates a new working representation of the method.
USE/ADVANTAGE - Relates to improved interpreter for optimising calls to methods whose sole purpose is to access value of private variable, modify private variable value or return constant value.
Dwg.5/5
Derwent Class: T01
International Patent Class (Main): G06F-009/06; G06F-009/44; G06F-009/45;
G06F-019/00

Source: World Wide Patent Files

File 256:SoftBase:Reviews,Companies&Prods. 85-1999/May
(c)1999 Info.Sources Inc
File 278:Microcomputer Software Guide 1999/Jun
(c) 1999 Reed Elsevier Inc.

Set	Items	Description
S1	1070	(STATIC? OR PREDETERMIN? OR PRESET? OR PRE() (DETERMIN?
OR -		SET? ? OR SETTING))
S2	177	INITIALI?
S3	1162	ARRAY?
S4	9690	(JAVA OR OBJECT()ORIENT? OR OOP OR OOPS OR VIRTUAL()MACHINE

S5 OR VM OR VMS OR JVM OR JVMS)
OR 579 (CODE? ? OR CODING OR COMPILER?) (10N) (OPTIMI? OR REDUC?
LESSEN? OR MINIMI? OR SHORTEN?)
S6 0 S1(S)S2(5N)S3(S)S4(S)S5
S7 0 S1(S)S2(5N)S3(S) (S4 OR S5)
S8 1 S1(5N)S2 AND S4
S9 2098 (CLINIT OR CLASS OR (MULTI OR MULTIPLE OR MANY OR
NUMEROUS-
) (2W) INSTRUCTION?)
S10 172 (SINGLE OR ONE OR SOLE) (2N) (EXPRESSION OR INSTRUCTION
OR C- OMMAND)
S11 7819 REPLAC? OR SUBSTITUT? OR TAKE(1W) PLACE OR EXCHANG?
S12 4 S9(5N)S11 AND (S4 OR S1(5N)S2)
S13 4 S12 NOT S8
? 8/3,K/1 (Item 1 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)1999 Info.Sources Inc. All rts. reserv.
00069882 DOCUMENT TYPE: Review
PRODUCT NAMES: C++ (800018); Standards (830218)
TITLE: Gavin Koch
AUTHOR: Schmidt, Doug
SOURCE: C++ Report, v6 n7 p56(4) Sep 1994
ISSN: 1040-6042
HOMEPAGE: http://www.sigs.com
RECORD TYPE: Review
REVIEW TYPE: Product Analysis
GRADE: Product Analysis, No Rating
REVISION DATE: 950301

...a set of quantities called 'implementation defined;' the programmer must establish, for example, a maximum number of arguments in a function call. Standardized order of initialization statics is not yet decided, but some responsibility by the programmer or the implementation seems inevitable. The group will mostly decide upon free-standing implementations for...
13/3,K/1 (Item 1 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)1999 Info.Sources Inc. All rts. reserv.

01616788 DOCUMENT TYPE: Product
PRODUCT NAME: CA-Visual Objects Lite (616788)
Computer Associates International Inc (081957)
1 Computer Associates Plaza
Islandia, NY 11788-7011 United States
TELEPHONE: (516) 342-5224

RECORD TYPE: Directory
CONTACT: Dana Williams, Product Inquiries
REVISION DATE: 970423
CA-Visual Objects Lite is an object -oriented application development system for Microsoft Windows. It combines system-level programming with an integrated development environment (IDE) that includes visual programming tools such as window, menu, database server and field editors. Access is provided through a set of replaceable database drivers and a class library that also encompasses GUI classes. The program features a fully object -oriented language that supports class creation, inheritance, behavior encapsulation and polymorphism while still providing support for CA-Clipper syntax. Programmers can create user-defined commands, functions ...

13/3,K/2 (Item 2 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)1999 Info.Sources Inc. All rts. reserv.
00106160 DOCUMENT TYPE: Review
PRODUCT NAMES: Clarion Professional Edition 4 (633321)

TITLE: Opening Windows Apps to Web
AUTHOR: Coffee, Peter
SOURCE: PC Week, v15 n10 p46(1) Mar 9, 1998
ISSN: 0740-1604
HOMEPAGE: http://www.pcweek.com

RECORD TYPE: Review
REVIEW TYPE: Review
GRADE: A

REVISION DATE: 981222

...entry and sorted record listings and requires an absolute minimum of effort or learning. Clarion's solid foundation for database application building now has full object -oriented programming built onto it. The Clarion language, an excellent high-level language, now has a class declaration facility that permits inheritance of data members and methods from a parent class. If the inheritance is not desired, programmers can still override a method name from the parent class and replace it with a specialized method that is deemed better to use with the newly defined subclass. Clarion's language also now permits multiple method definitions ...

13/3,K/3 (Item 3 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)1999 Info.Sources Inc. All rts. reserv.

00101734 DOCUMENT TYPE: Review

PRODUCT NAMES: Java (573744); HotJava (563251); HTTP (Hypertext Transport Protocol) (835307); Internet Inter-ORB Protocol (IIOP) (837032)

TITLE: A Web of Technologies
AUTHOR: Carlson, David, Ph.D.
SOURCE: Object Magazine, v6 n12 p18(2) Feb 1997
ISSN: 1055-3614
HOMEPAGE: http://www.sigs.com

RECORD TYPE: Review
REVIEW TYPE: Product Analysis
GRADE: Product Analysis, No Rating
REVISION DATE: 990521

PRODUCT NAMES: Java (
Sun Microsystems' Java and HotJava, JigSaw, Hypertext Transport Protocol
(HTTP), and Internet Inter-ORB Protocol (IIOP) are technologies and products mentioned in a discussion of the browser war....

...multiple formats, and the inability of a World Wide Web server's common gateway interface (CGI) to scale up to high-volume or advanced functionality. Java applets allow users to provide custom functions for protocols and content, but what is really needed is a fully extensible object -oriented Web browser. HotJava almost provides all the extensibility needed because it is totally developed in Java and has an all-purpose facility for extension of both protocol handlers and content handlers. Security should soon be enhanced with Java Development Kit 1.1.
The World Wide Web Consortium, W3C, is developing a next-generation, extensible Web server, JigSaw. It is fully implemented with Java and provides an all-purpose object -oriented design supporting minor and major extensions deployed as subclasses to its library, or more extreme class replacements that deploy particular interfaces. The OMG's contest for the best IIOP applications that can run from a Web client requires that the application show...

DESCRIPTORS: Internet Utilities; Java ; User Interfaces; Program Development Aids; Object Oriented Languages; Front Ends; Workstations; Public Networks; Thin Clients/Network Computers

13/3,K/4 (Item 4 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)1999 Info.Sources Inc. All rts. reserv.

00089421 DOCUMENT TYPE: Review

PRODUCT NAMES: ProductManager 3.0 (335266)

TITLE: IBM: On Target With PIM
AUTHOR: Halpern, Marc
SOURCE: Computer-Aided Engineering, v15 n3 p62(1) Mar 1996
ISSN: 0733-3536
HOMEPAGE: http://www.penton.com/cae/

RECORD TYPE: Review
REVIEW TYPE: Product Analysis
GRADE: Product Analysis, No Rating
REVISION DATE: 960730

...CAD integration not seen in other PIMS. With this integration, Product Manager's hierarchy can automatically reflect changes to Catia assemblies.
It also integrates an object -oriented lower CASE tool with a schema browser, which records changes whenever a user replaces a Product Manager class with custom code.

Source: Computer/product full text files on Dialog

File 275:COMPUTER DATABASE(TM) 1983-1999/JUN 30
(c) 1999 THE GALE GROUP
File 674:Computer News Fulltext 1989-1999/Jun W3
(c) 1999 IDG Communications
File 16:PROMT(R) 1972-1999/JUN 30
(c) 1999 THE GALE GROUP
File 15:ABI/INFORM(R) 1971-1999/Jun 30
(c) 1999 UMI
File 98:General Sci Abs/Full-Text 1984-1999/May
(c) 1999 The HW Wilson Co.
File 148:TRADE & INDUSTRY DATABASE 1976-1999/JUN 30
(c) 1999 THE GALE GROUP
File 636:NEWSLETTER DB(TM) 1987-1999/JUN 30
(c) 1999 THE GALE GROUP
File 624:MCGRAW-HILL PUBLICATIONS 1985-1999/JUN 29
(c) 1999 MCGRAW-HILL CO. INC
File 9:BUSINESS & INDUSTRY(R) JUL 1994-1999/JUN 30
(c) 1999 RESP. DB SVCS.
File 88:BUSINESS A.R.T.S. 1976-1999/JUN 30
(c) 1999 THE GALE GROUP
File 370:Science 1996-1999/May W2
(c) 1999 AAAS
File 612:Japan Economic Newswire(TM) 1984-1999/Jun 29
(c) 1999 Kyodo News
File 621:NEW PROD.ANNOU.(R) 1985-1999/JUN 30

(c) 1999 THE GALE GROUP
File 635:Business Dateline(R) 1985-1999/Jun 30
(c) 1999 UMI
File 484:PERIODICAL ABSTRACTS PLUSTEXT 1986-1999/JUN W2
(c) 1999 UMI
File 647:CMP Computer Fulltext 1988-1999/Jun W3
(c) 1999 CMP
File 813:PR Newswire 1987-1999/Apr 30
(c) 1999 PR Newswire Association Inc
File 810:Business Wire 1986-1999/Feb 28
(c) 1999 Business Wire

Set	Items	Description
S1	257660	(STATIC? OR PREDETERMIN? OR PRESET? OR PRE() (DETERMIN?
OR -		SET? ? OR SETTING))
S2	19995	INITIALI?
S3	472833	ARRAY?
S4	389465	(JAVA OR OBJECT()ORIENT? OR OOP OR OOPS OR
VIRTUAL()MACHINE		OR VM OR VMS OR JVM OR JVM(S)
S5	40420	(CODE? ? OR CODING OR COMPILER?) (10N) (OPTIMI? OR REDUC?
OR		LESSEN? OR MINIMI? OR SHORTEN?)
S6	0	S1(S)S2(5N)S3(S)S4(S)S5
S7	2	S1(S)S2(5N)S3(S) (S4 OR S5)
S8	2	RD (unique items)
S9	12	S1(5N)S2(S)S4
S10	12	S9 NOT S8
S11	9	RD (unique items)
S12	1299873	(CLINIT OR CLASS OR (MULTI OR MULTIPLE OR MANY OR
NUMEROUS-) (2N)INSTRUCTION?)
S13	27184	(SINGLE OR ONE OR SOLE) (2N) (EXPRESSION OR INSTRUCTION
OR C-		OMMAND)
S14	4536410	REPLAC? OR SUBSTITUT? OR TAKE(1W)PLACE OR EXCHANG?
S15	7	S12(S)S14(S)S13(S) (S4 OR S1(5N)S2)
S16	5	RD (unique items)
S17	88	S12(5N)S14(S)S4
S18	0	S17(S)S1(5N)S2
??t s8/3,k/all		
8/3,K/1		(Item 1 from file: 275)
DIALOG(R)File		275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP.		All rts. reserv.
01710223		SUPPLIER NUMBER: 16240084 (USE FORMAT 7 OR 9 FOR FULL
TEXT)		
We have mail.		(Letter to the Editor)
C/C++ Users Journal,		v12, n9, p109(8)
Sept, 1994		
DOCUMENT TYPE:		Letter to the Editor
ENGLISH		RECORD TYPE: FULLTEXT
WORD COUNT:		6272 LINE COUNT: 00520
ISSN:		1075-2838
LANGUAGE:		
...		embed an SCCS ID string in each source module such that it will
also be present in the		executable image. We originally did this by

initializing a static volatile const char array to the ID text string.
This variable was declared at the top of the module, outside of any functions. However, the compiler noticed that it wasn't used anywhere and optimized it out of existence. I thought the volatile type qualifier was supposed to prevent this sort of thing. We eventually found that a static function that returned a const char pointer to the ID text string was permitted to persist even though it was never called.
3. I learned...

8/3,K/2 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/INFORM(R)
(c) 1999 UMI. All rts. reserv.

00538111 91-12455
Nantucket Issues Challenge with Clipper 5.0
Fuller, Arthur
Computing Canada v17n5 PP: 26-27 Mar 1, 1991
ISSN: 0319-0161 JRNL CODE: CCD

...ABSTRACT: to Ashton-Tate's dBASE IV. One of the most important benefits of the new Clipper is its scoping abilities, which provide true local and static variables. Clipper 5.0 also introduces code blocks, an incredibly powerful addition to the language. A code block is a new data type, consisting of...

...general code, which is customized by code blocks. The most important new facet of Clipper 5.0 is its completely new implementation of arrays. An array may now be declared and initialized in one line. More important, any array element may contain an array, indefinitely. It will take the average Clipper programmer 200 hours or more to become comfortable in this new object - oriented programming environment, but once Clipper 5.0 stabilizes, it could sweep the high end of the xBASE community.
?t s11/3,k/all

11/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

02296680 SUPPLIER NUMBER: 54635645
Default Initialization of Built-in Types.
Schwartz, Robert Allan
C/C++ Users Journal, 17, 6, 71(1)
June, 1999

ISSN: 1075-2838 LANGUAGE: English RECORD TYPE: Abstract

ABSTRACT: C++ can profit from wrappers similar to those provided in Java

. The language guarantees that all objects are initialized as soon as they are created, but built-in types are initialized to zero by default for static storage classes while auto and heap variables are not initialized at all. A wrapper class acts as a constructor and initializes the built-in type...

11/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

02255907 SUPPLIER NUMBER: 21278232 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Smart cards and the Open Terminal Architecture.(programming multiple applications, terminals) (Technology Information) (Technical)

Conklin, Edward K.

Dr. Dobb's Journal, v23, n12, p70(6)
Dec, 1998

DOCUMENT TYPE: Technical ISSN: 1044-789X LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 3871 LINE COUNT: 00374

... controls whether socketed code is allowed to execute, and both terminal and card application code undergo stringent testing and certification.

Virtual Machine Architecture

The OTA Virtual Machine is a byte-addressed, two's complement, 32-bit machine, with 32-bit registers. It is based on a multiple-stack architecture, as is usual in Forth; see Figure 2. OTA was carefully designed to keep applications separate, and so the VM employs the so-called "Harvard Architecture," with separate code and data spaces. The data, return, and exception stacks, although technically in data RAM, are not in accessible memory space and can only be managed through specific tokens. Code memory is used only by the VM kernel and is not available to token programs (which are handled as data); thus, code memory can be considered to be in ROM. Initialized data space is preset at compile time, and will be instantiated in the target at run time. Uninitialized data is similar but preset to binary zero. These data spaces...

11/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

01711287 SUPPLIER NUMBER: 16003230 (USE FORMAT 7 OR 9 FOR FULL TEXT)
C/C++ Q & A. (Column)
DiLascia, Paul

Microsoft Systems Journal, v9, n7, p83(4)
July, 1994

DOCUMENT TYPE: Column ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 2219 LINE COUNT: 00189

... the compiler adds statis objects to its internal list, does it add them at the beginning or the end? Well, within a particula module, C++ initializes static objects in the order they appear--but what if your project contains several modules with static objects? In what order are all the objects initialized then? The answer is--surprise!--in no particular order at all: C++ makes no guarantee as to the order of initialization of static objects in different "translation units" (modules, for all practical purposes). In your MFC example, suppose some other static object in another module has a construtor....

...before the App? The return value from AfxGetApp will be NULL becuse the CWinApp constructor hasn't run yet and afxCurrenWinApp hasn't been set.

Oops .

Figure 1 shows how easy it is to write a program that manifests this problem of unordered initializations, to coin a phrase. (Kinda has a...

11/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

01707457 SUPPLIER NUMBER: 16286126 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Quincy's translator and the C++ library.

Stevens, Al

Dr. Dobb's Journal, v19, n13, p109(7)
Nov, 1994

ISSN: 1044-789X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT;
ABSTRACT

WORD COUNT: 2830 LINE COUNT: 00223

... lexical scanner.

Quincy interprets the token stream, which encodes source code.

While some interpreters compile the tokens into a pseudocode that, when interpreted, implements a virtual -machine architecture, Quincy does not. Its translation consists of scanning the source code into tokens; building and initializing the global and static declarations; and resolving symbol references to global, local, and argument identifiers. However, the translator must first establish the run-time environment and call the preprocessor...

11/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

01532290 SUPPLIER NUMBER: 12522624 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Code of practice. (the technicalities of application development using Cobol on the VAX) (FilovAX) (Technical)
Shelton, Peter
DEC User, p43(2)
July, 1992
DOCUMENT TYPE: Technical ISSN: 0263-6530 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1694 LINE COUNT: 00126

...ABSTRACT: to debug. Cobol can be used with database systems, transaction processing systems and the DECforms forms product. The 'value' clause should be used only for initializing static data whose value will never change. Binary fields should be utilized for arithmetic. Cobol
Report Writer, a standard part of VMS Cobol, should be used for reports.
Be careful when coding disk I/O and terminal I/O as these areas are time intensive. All source...

11/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

01366988 SUPPLIER NUMBER: 08697746 (USE FORMAT 7 OR 9 FOR FULL TEXT)
C++ Release 2.0. (enhancements)
Eckel, Bruce
Dr. Dobb's Journal, v15, n8, p96(2)
August, 1990
ISSN: 1044-789X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1362 LINE COUNT: 00103

ABSTRACT: AT&T's C++ object -oriented programming language creator Bjarne Stroustrup and his team introduced major enhancements in version 2.0, but Stroustrup later found and resolved some small problems in...

...committee. C++ 2.0 enhancements include support for abstract classes; copying and initialization; class-by-class overloading of operators new and delete; const, volatile and static member functions; sophisticated initialization ; and pointers to members. 2.1 enhancements include classes, enumerations and typedefs defined in classes not local to them, inheritance of pure virtual functions as...

11/3,K/7 (Item 1 from file: 16)
DIALOG(R)File 16:PROMT(R)
(c) 1999 THE GALE GROUP. All rts. reserv.

04151230
Tartan Ada Development System Available For Ariel 1.1-BOPS VMEbus DSP Board
News Release October 14, 1992 p. 1

... military applications. The system provides facilities for building, testing, and debugging large-scale Ada programs. The development system may be hosted on either a VAX/ VMS or Sparc/SunOS system, includes an optimizing Ada compiler, Ada librarian, TMS320C40 simulator, window-oriented symbolic debugger, and a linker. It also includes routines for...

... on one of the most critical concerns in DSP system design -- execution speed. Ada-specific optimizations include constraint and overflow check elimination, data packing, and static aggregates initialization
320C40-specific optimizations include support for the DSP's auto increment, repeat block, delayed branch, and loop folding instructions. The compiler provides five optimrztion levels...

11/3,K/8 (Item 2 from file: 16)
DIALOG(R)File 16:PROMT(R)
(c) 1999 THE GALE GROUP. All rts. reserv.

01709894
SIERRA SYSTEMS ANNOUNCES HIGH PERFORMANCE 68020/68881 C CROSS DEVELOPMENT SYSTEM .

NEWS RELEASE May 13, 1987 p. 11

...floating point emulation, librarian, command driver, and object code utilities. Hosts currently include the IBM PC/AT under MS-DOS and the DEC VAX under VMS and UNIX. The user interface is identical accross all hosts. The package has been designed specifically for embedded systems applications, providing features like ROMable, position independent and re-entrant code, run-time initialization of static data, and support

for resident libraries and highly fragmented address spaces. The PC/AT hosted version of Sierra C runs at speeds comparable to VAX...

11/3,K/9 (Item 1 from file: 621)
DIALOG(R)File 621:NEW PROD.ANNOU.(R)
(c) 1999 THE GALE GROUP. All rts. reserv.

00184106 00184106

OSAYS Offers Sierra C Cross Development Package For Motorola 68020

News Release
DATELINE: Waltham, MA March 21, 1988 WORD COUNT: 523

...explicitly designed for embedded systems development. The following features are built-in:
* ROMable code generation
* Position independent code generation
* Re-entrant code generation
* Run-time initialization of static data
* Support for resident libraries
OSAYS Sierra C support the proposed ANSI C standard new keywords volatile and const.
OSAYS, Inc., located in Waltham, MA offers over 120 tools for software development using micro, mini, and mainframe systems running UNIX, VMS , MS-DOS and other environments. OSAYS' products, many developed through OEM, VAR, and other exclusive distribution relationships, are available for native and cross-development with...

16/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:COMPUTER DATABASE(TM)
(c) 1999 THE GALE GROUP. All rts. reserv.

01406125 SUPPLIER NUMBER: 11361745
New OS at core of IBM, Apple pact. (International Business Machines Corp., Apple Computer Inc., operating systems) Gillooly, Brian; Rohm, Wendy Goldman
Computer Reseller News, n440, p1(2)
Sept 30, 1991
ISSN: 0893-8377 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: The technology-exchange agreement between noted rivals Apple Computer Inc and IBM Corp is expected to be signed on Oct 2, 1991. Apple and IBM will announce the creation of a separate company, possibly called AIM for Apple, IBM and Motorola Inc, that will develop and market object - oriented operating-system software. Apple is licensing IBM's single - chip reduced-instruction -set computer (RISC) technology and OEM UNIX systems

to create its own RISC-based systems. Apple plans to place the Macintosh application interface onto UNIX...

...Macintosh productivity applications. The operating system software has been code-named Pink and will run not only on RISC-based workstations but Intel 80X86 computers. Object -oriented icon-based class libraries will be incorporated into the operating system and licensed to third-party developers. Since the cooperative agreement was first discussed, 150 Apple and IBM...

16/3,K/2 (Item 1 from file: 674)
DIALOG(R)File 674:Computer News Fulltext
(c) 1999 IDG Communications. All rts. reserv.

068697
Novell delivers
Lab test shows new features and bundled extras make NetWare 5 a compelling upgrade.

Byline: James E. Gaskin
Journal: Network World Page Number: 14
Publication Date: September 07, 1998
Word Count: 908 Line Count: 84

Text:
...and found its new features - pure IP support, robust memory management, enhanced directory services and an eye-catching graphical administration interface - make it a World Class product.Novell has rewritten NetWare to use pure TCP/IP and has even made it the default network protocol, dethroning Novell's own IPX. The...

... Host Configuration Protocol tools that load automatically and are controlled through NWAdmin, Novell's NDS administrative interface. Other additions include the Service Locator Protocol, which replaces most of IPX's chatty service advertisement broadcasts, and NetWare Storage Services. Large network volumes used to be slow to mount and prepare during server...

... is the easy upgrade to Client32 software for Windows 95, 98 and NT users. Simply copy some files to a network directory and add a single

command to appropriate logon scripts, and every user's files will be brought up to date. NetWare 5 includes a ZENworks starter pack, which offers automated...

...The one nit we found on the server side is the new ConsoleOne management interface. Sure, it's pretty, and it's written in 100% Java . But it's slow, even when running on a workstation, and it's immature compared with the long-time administration utility, NetWare Administrator, which is...

16/3,K/3 (Item 2 from file: 674)
DIALOG(R)File 674:Computer News Fulltext
(c) 1999 IDG Communications. All rts. reserv.

039139
The OOP SURVIVAL GUIDE
OOP SURVIVAL GUIDE
Replacing THE 1s and Os with objects is easy. Mastering the mind-set for object-oriented programming is hard. Here to smooth your way is The OOP Survival Guide
Byline: Carl A. Argila
Journal: Computerworld Page Number: 89
Publication Date: August 22, 1994
Word Count: 1537 Line Count: 140

Text:
...critical to get the model right. The following story illustrates why:
A European country replaced a legacy system designed to track government benefits with an object - oriented application. The programmers identified object classes with names such as ``pension`` and ``beneficiary``. The system worked well until it was time to update the ``legislative rules`` embedded throughout. Programmers had never created a legislative rules object class . So instead of issuing one command to perform a global update, they had to change each one step by step. That defeats the point of OOP .
Specify a time-frame and stick to it
THE PROJECT DELIVERY DATE never slips. Period. If it does, the technology you are trying to promote...

DIALOG(R)File 621:NEW PROD.ANNOU. (R)
(c) 1999 THE GALE GROUP. All rts. reserv.

00737302 00737740

POET First to Offer Java Developers Standard Interface to Full-Featured Object Database; ODMG Java Tight Binding Added to Advanced POET Universal Object Server Only Database to Achieve Completely Transparent Java, C++ Object Interchange.

Business Wire
DATELINE: SAN MATEO, Calif. Feb 18, 1997 WORD COUNT: 1506

SAN MATEO, Calif.--(BUSINESS WIRE)--Feb. 18, 1997--POET Software today announced it is delivering a pre-release version of its 100-percent ODMG-compliant Java tight binding, making POET the only full-featured object database to offer Java developers this industry-standard interface.

The addition of the ODMG Java tight binding makes POET's Universal Object Server the only object database offering direct, transparent access to all objects from either Java or C++ applications.

POET's Java solution goes well beyond non-scalable, single-user, persistent object storage to give developers access to the full power of the POET Universal Object Server's advanced functionality for their large-scale Java applications, including multiple database transactions, multi-user access, queries, replication and publish/subscribe.

The POET Universal Object Server family now provides C++ and Java SDKs, allowing developers to write applications that exchange C++ and Java objects completely transparently. In the next version, POET will enable Java and C++ applications to access data from relational databases, an important consideration for any organization with large volumes of legacy data.

"A number of database vendors are claiming to offer a Java interface and language interoperability, but only POET is actually delivering a full-featured object database with ODMG Java tight binding and transparent interoperability to its customers," said Dirk Bartels, president and CEO of POET Software.

"POET gives developers the fastest, easiest path for migrating from C++ to Java applications, and also supports Visual Basic applications and ActiveX controls."

"We are pleased to have POET's object database technology supporting Java ," said Rick Cattell, distinguished engineer at JavaSoft. "Java has become the key to the rapid development and deployment of a new generation of applications on the Internet. Now, with Java support, we expect that POET's Universal Object Server will become an important component of this trend."

"POET's Java software development kit is a significant addition to POET's Universal Object Server suite," said Denice Gibson, senior

vice president of Novell, Inc. "POET's new Java SDK provides us with a broad range of options for delivering enhanced value in our upcoming releases of various Novell products. One of the areas we will be using POET technology is to improve the installation of Novell products."

POET Java 1.0

POET Java 1.0 is the first implementation of the Java tight binding as defined by the work-in-progress draft of Chapter 7 of the ODMG Release 2.0 standard. POET is a voting member of the Object Database Management Group, and has been actively involved in defining this standard.

POET's Java tight binding creates a single, unified object-type system shared by the Java language and the POET object database. The programmer perceives the binding as an extension to the development language.

The syntax used to create, delete, identify, reference get/set field values and invoke methods on a persistent object is no different from that used for objects of shorter lifetimes, allowing a single expression to freely intermix references to persistent and transient objects.

The full functionality of the Object Query Language (OQL) is available through the POET Java tight binding. This functionality can be used through query methods on class Collection or through queries using a standalone OQLQuery class. Persistence is achieved by reachability as defined in the ODMG standard.

All objects reachable from database root objects are retrieved from or stored to the database, and objects no longer referenced are removed from the database automatically.

POET Java 1.0 takes full advantage of POET's advanced server functionality, including automatic schema evolution; direct, transparent access to C++ objects; binary large objects and streaming; object-level granularity; and explicit object locking.

Further, POET Java 1.0 can use the local POET database engine to store Java objects in a local POET database that can be accessed and used by any other POET application or copied to a different machine running a different operating system. The physical database is 100-percent binary compatible across all platforms and programming languages.

The POET Java SDK contains, in addition to POET Java 1.0, POET's Java generic "loose" language binding that is indispensable for developing generic database applications in Java, like a Web browser, that must be able to access any POET database regardless of its underlying schema. It also allows writing interfaces to existing C++ applications without reimplementing these persistent classes in Java.

As with all POET language bindings, the Java generic binding supports local storage and full client-server connectivity, and can be used as an ultra-thin pure Java client in conjunction with an intermediate object request broker.

POET Java Strategy

The Java language has burst upon the scene with remarkable speed, and some estimates put the number of Java developers at more than 300,000. According to a 1996 report by Forrester Research, 62 percent of companies with active Internet initiatives in the Fortune 1,000 are already using Java, and 42 percent are convinced that Java will be key to their Internet computing strategy within the year.

Java fixes many of the major shortcomings of C++ -- memory management, pointers, built-in multithreading and standardized class libraries among others -- and is the first object-oriented programming language to offer truly hardware-independent programming.

POET believes that Java is a tremendous advance in object-oriented programming, and is embarking on an evolutionary approach to providing Java support for the POET Universal Object Server, one that will parallel the evolution of Java as it becomes a more mature language.

The first release of the POET Java tight binding contains a Java client and relies on the tested and proven fourth-generation POET kernel written in C++ for the local client database engine. The POET ODBMS is persistent objects while supporting Java's platform independence.

Targeted for the end of 1997, POET will introduce a pure Java client supporting full connectivity to any POET database server without the need for the POET C++ kernel on the client. Client applications will be able to be downloaded and run on any platform that supports Java, opening the door to a wide range of sophisticated Internet/intranet applications requiring powerful database access.

In addition, POET will offer an "ultra-thin" client that uses an intermediate object request broker in a three-tier architecture for access to a POET server, enabling the use of persistent objects in Java applets with minimum memory requirements and greatly reduced download times.

As the Java language matures, POET will implement a remote Java client database engine that will allow developers to write pure Java applications using local storage to keep objects stored on the client side. This can be used for data replication, check-in/check-out of objects...

...storing client data like user information and configuration data. Massive data storage can still be achieved from the POET Universal Object Server through the pure Java client implementation.

In addition to language binding and local storage, POET will use the advantages of Java in the POET server itself. The hardware-independent nature of Java makes it a perfect fit for use as an execution environment in the database server, enabling active methods stored in the database to be executed on the server side. Because Java byte code is totally machine independent, these

E.I.C. Search History for Application No. 09/055,947

methods can be executed on every platform supporting a Java virtual machine .

"Using this capability, a Java developer will be able to implement an entire application based on persistent objects stored in the database, and will gain powerful features such as dynamic...
...main method. The code is stored in the database, and can be executed either on the client or server side."

Pricing and Availability

The POET Java Software Development Kit, available today for \$499, contains a beta release of the POET Java 1.0 ODMG-compliant tight binding, and the 1.0 release of the POET Java generic binding. The full product release will be available in June for \$499. An evaluation version is free and available now for download from the...

16/3,K/5 (Item 2 from file: 621)
DIALOG(R)File 621:NEW PROD.ANNOU.(R)
(c) 1999 THE GALE GROUP. All rts. reserv.

00164532 00164532

IBM'S RESEARCH DIVISION READIES 512-UNIT PARALLEL PROCESSOR

DATELINE: YORKTOWN HEIGHTS, NY September, 1987 WORD COUNT: 1813

...Years have advances in chip design, switching speed, network architecture, and programming made highly parallel processors such as the RP3 feasible.
The RP3 is a multiple instruction , multiple data (MIMD) machine, in which each processor is able to operate independently. This freedom sets the RP3 apart from single instruction , multiple data (SIMD) parallel processors, in which all processors are the same as those used in IBM's RT PC (based on RISC -- reduced instruction...

...components. Rather than spend time custom-designing ever more compact parts in order to bring down the RP3's size, or interrupting design work to replace last year's parts with this year's, the designers realized that sticking with readily available hardware would simplify their job. In the words of...
...do precisely that.
To find out how a program will run on the RP3, software developers have used an experimental multi-processor operating system called VM /EPEX (virtual machine /environment for parallel execution). Run on an IBM System/370, VM /EPEX emulates the RP3's programming environment. In this way, researchers have already modified more than 40 applications programs for RP3-like parallelism.